

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Robert Miller

Appeal No. _____
Application No. 09/732,189

AMENDED APPEAL BRIEF

Applicant: Robert Miller Art Unit: 2157
Serial No.: 09/732,189 Examiner: Hussein A. El-Chanti
Filed: December 7, 2000
For: PEER PROTOCOL STATUS QUERY IN CLUSTERED COMPUTER SYSTEM

Page 1 of 15
Serial No. 09/732,189
Amended Appeal Brief dated February 2, 2007
IBM Docket ROC920000125US1
WH&E IBM/151

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's invention is generally directed to an apparatus, program product and method that locally track protocol progress information within each member of a group of a clustered computer system, for the purpose of identifying at least one problematic member of the group. By locally tracking such information, any member of the group may be directed to provide such information on demand in response to a query directed to such member. As a consequence, if there are N members of a group, and one is problematic, the probability of successfully obtaining the protocol progress information via a query directed to an arbitrary group member is at worst $(N-1)/N$ (assuming a problematic member is incapable of responding to a query). Typically, in such a situation, at most two requests would be needed to obtain the necessary information.

A clustered computer system is a computer system where multiple computers, or nodes, are networked together to present a single system image (Application, p. 1, lines 10-14). Clusters perform tasks through the performance of jobs running on each node, which may be logically organized together into a "group" to perform collective tasks, where each affiliated job is referred to as a "member" of the group (Application, p. 2, lines 1-5). Joint operations performed by members of a group are referred to as "protocols", and in some clustered systems these protocols are implemented as "peer protocols", where "all members receive a message and each member is required to locally determine how to process the protocol and return an acknowledgment indicating whether the message was successfully processed by that member" (Application, p. 2, lines 13-16).

Applicant's invention addresses the problem that arises in clustered computer systems whereby the failure of a member of a group to respond quickly (or at all) to a protocol request will typically slow down the rest of the members, often because each member is required to wait for acknowledgments from all other members before proceeding on to a next protocol (Application, p. 2, lines 19-30). Traditionally, when one member is stuck or slow, all members appear to be stuck or slow as well, and as a result, it becomes difficult to isolate the problem to a particular member.

Embodiments of the invention address this problem by requiring each member to locally track protocol progress information, and then configuring each member to respond to a query by providing its locally tracked protocol progress information (Application, p. 4, lines 1-12).

In some embodiments, each member participating in a protocol is required to send an ACK message to each other member in the group when the member has completed processing of the protocol. In this regard, protocol progress information may be locally tracked by tracking, within each member, a current ACK round for such member, as well as the last ACK round received from each other member (Application, p. 10, lines 3-17). By doing so, identification of a slow or stuck member is often trivial from a review of the protocol progress information, as a slow or stuck member will have a last ACK round received value that lags that of the other members (Application, p. 10, lines 18-25).

An additional feature supported by embodiments of the invention is the ability to monitor, in a member of a cluster group, for receipt of a query message while that member is waiting on a resource, and providing protocol status information in response to receipt of the query message (Application, p. 4, lines 18-22). This feature may be implemented, for example, by placing protocol messages on a message queue in a member, and continuing to monitor for query messages while a protocol is being processed and waiting on a resource (Application, p. 12, line 14-p. 13, line 12, Fig. 6). The protocol being waited upon may be a peer protocol, or a local protocol, e.g., a protocol waiting on a lock or a creation of a new job (Application, p. 11, lines 3-9).

Among the claims on appeal, claims 1, 14, 22, 23, and 25 are independent. Support for the claimed subject matter in claims 1, 14, 22, and 23 may be found, for example, in the Application, at p. 2, lines 13-16, p. 4, lines 1-12, p.7, lines 1-8, and p. 10, lines 6-20, and in Fig. 4 as filed. Support for the claimed subject matter in claim 25 may be found, for example, in the Application, at p. 4, lines 18-22, p. 10, line 29 to p. 11, line 23, and p. 12, line 14 to p. 13, line 12, and in Figs. 4 and 6 as filed.

Specific support for the claimed subject matter for the independent claims as a whole has been provided above. However, a direct mapping of the aforementioned discussion to the

individual independent claims, as requested by the Examiner in the Notification of Non-Compliant Appeal Brief dated January 24, 2007, is presented below:

Independent Claim 1

A method of determining a status of a peer protocol initiated on a plurality of members of a group in a clustered computer system (Application, p. 4, lines 2-13), the method comprising:

(a) locally tracking protocol progress information for a peer protocol within each of a plurality of members collectively managed as a group by a clustered computer system (Application, p. 4, lines 13-15, p. 11, line 28 - p. 12, line 17, Fig. 5), wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol (Application, p. 6, line 28 - p. 7, line 8); and

(b) responding to a query directed to a selected member of the group by providing the protocol progress information locally tracked by the selected member, wherein the query comprises a request for the protocol progress information (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6).

Independent Claim 14

An apparatus, comprising:

(a) a memory (Fig. 2, block 14); and

(b) a program (Fig. 3) resident in the memory, the program configured to determine a status of a peer protocol initiated on a plurality of members that are collectively managed as a group by a clustered computer system (Application, p. 4, lines 2-13) by locally tracking protocol progress information within at least one member of the group (Application, p. 4, lines 13-15, p. 11, line 28 - p. 12, line 17, Fig. 5), and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6), wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in

association with locally processing the peer protocol (Application, p. 6, line 28 - p. 7, line 8), and wherein the query comprises a request for the protocol progress information (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6).

Independent Claim 22

A clustered computer system (Fig. 1, block 2), comprising:

- (a) a plurality of nodes (Fig. 1, block 10) coupled to one another over a network (Fig. 1, blocks 4, 6, 8);
- (b) a plurality of members (Fig. 1, block J1-J7) collectively managed as a group by the clustered computer system and configured to be executed by at least one of the plurality of nodes; and
- (c) a program (Fig. 3) configured to be executed by at least one of the plurality of nodes to determine a status of a peer protocol initiated on the plurality of members (Application, p. 4, lines 2-13) by locally tracking protocol progress information within at least one member of the group (Application, p. 4, lines 13-15, p. 11, line 28 - p. 12, line 17, Fig. 5), and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6), wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol (Application, p. 6, line 28 - p. 7, line 8), and wherein the query comprises a request for the protocol progress information (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6).

Independent Claim 23

A program product, comprising:

- (a) a program (Fig. 3) configured to determine a status of a peer protocol initiated on a plurality of members that are collectively managed as a group by a clustered computer system (Application, p. 4, lines 2-13) by locally tracking protocol progress

information within at least one member of the group (Application, p. 4, lines 13-15, p. 11, line 28 - p. 12, line 17, Fig. 5), and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6), wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol (Application, p. 6, line 28 - p. 7, line 8), and wherein the query comprises a request for the protocol progress information (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6); and

(b) a signal bearing medium bearing the program.

Independent Claim 25

An apparatus, comprising:

(a) a memory (Fig. 2, block 14); and

(b) a program (Fig. 3), resident in the memory, the program configured to monitor for receipt of a query message that requests protocol status information by a member among a plurality of members that are collectively managed as a group by a clustered computer system while a current protocol for the member is waiting on a resource, (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6) the program further configured to output protocol status information in response to receipt of the query message (Application, p. 4, lines 16-17, p. 12, line 18 - p. 13, line 17, Fig. 6).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-26 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,138,251 to Murphy (*Murphy*).

VII. ARGUMENT

Applicant respectfully submits that the Examiner's rejections of claims 1-26 are not supported on the record, and should be reversed. Anticipation of a claim under 35 U.S.C. §102

requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros., Inc. v. Union Oil Co., 2 USPQ2d 1051, 1053 (Fed. Cir. 1987), *quoted in In re Robertson*, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999). Absent express description, anticipation under inherency requires extrinsic evidence that makes it clear that "the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill." Continental Can Co. v. Monsanto Co., 20 USPQ2d 1746, 1749 (Fed. Cir. 1991), *quoted in In re Robertson* at 1951. "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." Continental Can at 1749, *quoted in In re Robertson* at 1951.

Applicant respectfully submits that *Murphy* does not disclose the various features recited in claims 1-26, and as such, the rejections thereof should be reversed. Applicant will hereinafter address the various claims that are the subject of the Examiner's rejection in order, starting with the independent claims, and then addressing additional dependent claims reciting additional subject matter that is distinguishable from *Murphy*. In some cases, specific discussions of particular claims are not made in the interests of streamlining the appeal. The omission of a discussion with respect to any particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejections applied to other claims pending in the appeal.

Independent Claim 1

Claim 1 recites a method of determining a status of a peer protocol initiated on a plurality of members of a group in a clustered computer system. The method includes locally tracking protocol progress information for a peer protocol within each of a plurality of members collectively managed as a group by a clustered computer system, wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol, and responding to a query directed to a selected member of the group by providing the protocol

progress information locally tracked by the selected member, wherein the query comprises a request for the protocol progress information.

Claim 1 notably recites a number of features that focus the claim into a clustered computer environment. Specifically, claim 1 recites that a clustered computer system collectively manages a plurality of members as a group. In addition, the claim recites that protocol progress information, which is locally tracked by each member of the group, and which is provided in response to a query directed to a selected member of the group, is for a peer protocol of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol.

"Groups" and "members" are well known entities utilized in clustered computer systems. The Application, at page 1, line 28 to page 2, line 5, defines members and groups as follows:

Clusters typically handle computer tasks through the performance of “jobs” or “processes” within individual nodes. In some instances, jobs being performed by different nodes cooperate with one another to handle a computer task. Such cooperative jobs are typically capable of communicating with one another, and are typically managed in a cluster using a logical entity known as a “group.” A group is typically assigned some form of identifier, and each job in the group is tagged with that identifier to indicate its membership in the group.

Members are therefore jobs or processes executing on a node in a cluster, while a group is an entity that enables the member jobs to be collectively managed by a cluster. As such, within clustering environments, membership in a clustering group distinguishes group members from arbitrarily selected sets of entities that happen to interact with one another in a distributed computer system.

The Examiner apparently relies on col. 3, lines 32-36 of *Murphy* for allegedly disclosing a plurality of members of a group. The cited passage, however, merely refers to nodes, which are described as independent client/server computers. There is nothing in the passage, or elsewhere in the reference, that discusses the concept of a group of collectively managed member jobs running on different nodes of a clustered computer system. Accordingly, *Murphy* fails to anticipate this recited feature of claim 1.

With regard to the concept of locally tracking the progress of a peer protocol, the Examiner apparently relies on col. 4, lines 50-55 and col. 6, line 54 to col. 7, line 7 of *Murphy* for allegedly disclosing determining the status of a peer protocol via local tracking of protocol progress information within each member of a group, relying specifically on the teaching in *Murphy* of a client tracking protocol progress when downloading an object using an object reference. However, it is clear from these passages, and from elsewhere in *Murphy*, that the tracking of object references in clients and servers in *Murphy* does not disclose the concept of tracking the progress of a peer protocol, which has been defined in claim 1 as being a type where each member of a group receives as message associated with the protocol and returns an acknowledgment in association with locally processing the protocol. Nor does *Murphy* disclose any progress tracking that is performed by each member of a group that is collectively managed by a clustered computer system, as is also required by claim 1.

Instead, *Murphy* discloses an object reference tracking system between individual clients and servers in a clustered computer system. Generally, when a client requests an object reference, counts are incremented on each of the server and the client, and when a client discards an object reference, counts are decremented on each of the server and the client. As noted above, however, the clients and servers are not collectively managed via a logical entity that is analogous to a group. Moreover, the tracking of object reference counts occurs via specific interaction between a server and individual clients (rather than the server and all clients), to the extent that some clients are incapable of tracking operations occurring on other clients. One aspect of the invention recited in claim 1 is that each member of a group locally tracks the progress of a protocol initiated on the plurality of members, a feature that is not disclosed in *Murphy*. In contrast, in *Murphy* the majority of the operations that are performed are interactions between a server and an individual client, or between two clients, and consequently another client not involved in the interaction has no mechanism for tracking the interactions associated with other clients (see, e.g., the operations discussed in connection with Figs. 2A, 2B, 2C and 2E.)

In fact, the only operation in *Murphy* that appears to incorporate more than simple point-to-point messages between two nodes is illustrated in Fig. 2D and described in the aforementioned cited passage at col. 6, line 54 to col. 7, line 7. This operation, however, still

does not conform to the recited definition of a "peer protocol," where members of a group receive a message associated with a protocol and send an acknowledgment in connection with locally processing the protocol. Even though a server and multiple clients are involved, if any additional clients are present in the system, those additional clients will still not participate in the operation. As such, not even this operation incorporates a message received by each member of a group along with an acknowledgment from each member, as required by claim 1. *Murphy* therefore fails to anticipate the concept of locally tracking protocol progress information for a peer protocol within each of a plurality of members collectively managed as a group, as is also required by claim 1.

With respect to the concept of responding to a query directed to a selected member of the group by providing the locally tracked protocol progress information, Applicants have clarified that such a query comprises a request for the protocol progress information.

In rejecting this feature of claim 1, the Examiner cites col. 7, lines 9-32 of *Murphy*, arguing that the passage discloses that "[c]lient A sends a foreign ref count to client B, client B will determine that B already has the reference and respond to A by sending a DEC message to node A" (Final Office Action, p. 3).

The cited passage, however, discloses that client A may send an object reference, not a foreign reference count, to client B. In turn, if client B already has the object reference, client B merely sends a "DEC" message back to client A to inform the client to decrement its foreign reference count to prevent client B from being counted multiple times in client A's foreign reference count.

Moreover, there is no disclosure in *Murphy* that the transmission of an object reference constitutes anything analogous to a "request for . . . protocol progress information" as required by claim 1. Considering that claim 1 requires that the locally tracked protocol progress information be provided by the recipient of a request for such information, Applicant is unsure as to how *Murphy*'s communication of a "DEC" message in response to being sent an object reference could be found to be analogous to a response to a request for operation.

Indeed, it appears that the only information communicated between nodes in *Murphy* are object references and requests therefor, INC and DEC messages, which request another node to

increment or decrement their own object reference counts, and ACK messages, which acknowledge INC and DEC messages. While INC and DEC messages are commands to increment or decrement reference counts, which the Examiner apparently considers to correspond to protocol progress information, it does not appear that the counts themselves are ever transmitted in *Murphy*. As such, *Murphy* fails to anticipate the concept of responding to a query directed to a selected member of a group by providing protocol progress information locally tracked by that member, as is also required by claim 1.

As a final matter, the Examiner did not address any of the arguments presented by Applicant in the prior Amendment and Response filed July 1, 2005. Instead, the Examiner merely indicated that the arguments made therein were moot in view of new grounds of rejection – despite the fact that the rejection was not in fact new. Accordingly, Applicant has been unable to ascertain the Examiner's position as to any of the arguments previously made and now reiterated herein.

Applicant therefore respectfully submits that claim 1 is novel over *Murphy*, and the rejection of claim 1 should be reversed. Moreover, given that *Murphy* additionally fails to suggest the aforementioned features of claim 1, and that the Examiner has presented no objective evidence as to any motivation in the art to modify *Murphy* to incorporate any of such features, Applicant submits that claim 1 is also non-obvious over *Murphy*. Reversal of the Examiner's rejections, and allowance of independent claim 1, are therefore respectfully requested.

Independent Claims 14, 22 and 23

Next with respect to the rejections of independent claims 14, 22 and 23, each of these claims recites, similarly to claim 1, the concepts of determining the status of a peer protocol initiated on a plurality of members that are collectively managed as a group, locally tracking protocol progress information a member of the group, and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member. As discussed above in connection with claim 1, none of these features are disclosed or suggested by *Murphy*. Applicant therefore submits that each such claim is novel and non-

obvious over *Murphy*. Reversal of the Examiner's rejections, and allowance of independent claims 14, 22 and 23, are therefore respectfully requested.

Independent Claim 25

Next with respect the rejection of independent claim 25, this claim recites an apparatus that includes a memory, and a program resident in the memory. The program is configured to monitor for receipt of a query message that requests protocol status information by a member among a plurality of members that are collectively managed as a group by a clustered computer system while a current protocol for the member is waiting on a resource, and to output protocol status information in response to receipt of the query message.

As noted above in connection with claim 1, the concepts of a query message that requests protocol status information, and a protocol being performed by a plurality of members of a group that are collectively managed by a clustered computer system are neither disclosed nor suggested by *Murphy*. Furthermore, it should be noted that claim 25 additionally recites the concept of outputting protocol status information in response to receipt of a query message requesting such information, combined with monitoring for receipt of the query while waiting on a resource. The Examiner cites col. 7, lines 9-32 of *Murphy*, however, as noted above, this passage fails to disclose any transmission of information that could be analogized to protocol status information. Instead, the passage discloses the transmission of object references and requests therefor, INC and DEC messages, which are merely commands to increment or decrement reference counts, and ACK messages, which acknowledge receipt of INC and DEC messages. Moreover, there is nothing in the passage that addresses the concept of monitoring for receipt of a query while waiting on a resource.

As such, Applicant submits that claim 25 is novel and non-obvious over *Murphy*. Reversal of the Examiner's rejection, and allowance of independent claim 25, are therefore respectfully requested.

Dependent Claims 2 and 15

Claims 2 and 15 respectively depend from claims 1 and 14, and additionally recite tracking, within a first member of a group, ACK messages directed to the first member by each other member of the group. In rejecting these claims, the Examiner relies on *Murphy*, col. 7, lines 1-20. The cited passage, however, merely discloses the use of ACK messages to acknowledge INC and DEC messages. There is no disclosure in the reference of the tracking of ACK messages sent to one member by each other member of a group. Accordingly, claims 2 and 15 are novel over *Murphy* based upon this additional feature, and the rejections thereof should be reversed.

Dependent Claims 3-4 and 16

Claims 3 and 16 respectively depend from claims 1 and 14, and additionally recite tracking, within a first member of a group, a current ACK round associated with a current peer protocol being processed, and a last ACK round received parameter associated with each other member of the group. Claim 4 depends from claim 3 and recites updating a current ACK round for a first member in response to receiving ACK messages from all other members of a group. In rejecting these claims, the Examiner again relies on *Murphy*, col. 7, lines 1-20. The cited passage, however, merely discloses the use of ACK messages to acknowledge INC and DEC messages. There is no disclosure in the reference of the concept of ACK rounds, much less the specific tracking steps recited in these claims. Accordingly, claims 3-4 and 16 are novel over *Murphy* based upon this additional feature, and the rejections thereof should be reversed.

Dependent Claim 5

Claim 5 is not separately argued.

Dependent Claims 6-8 and 17-19

Claims 6 and 17 respectively depend from claims 1 and 14, and additionally recite waiting on a resource required by a protocol being processed on a selected member, and monitoring for receipt of a query while waiting on the resource. As discussed above in

connection with claim 25, these concepts are not anticipated by *Murphy*. Accordingly, these claims are novel over *Murphy* for the same reasons set forth above in connection with claim 25.

In addition, claims 7 and 18 clarify that the protocol is a peer protocol, while claims 8 and 19 clarify that the protocol is a local protocol. In rejecting these claims, the Examiner once again cites *Murphy*, col. 7, lines 1-20. The cited passage, however, merely discloses the use of ACK messages to acknowledge INC and DEC messages. There is no disclosure in the reference of waiting on resources required by a peer protocol or a local protocol, and as such, *Murphy* cannot be found to anticipate these claims. Accordingly, claims 6-8 and 17-19 are novel over *Murphy* and the rejections thereof should be reversed.

Dependent Claims 9-11 and 20

Claims 9-11 and 20 are not separately argued.

Dependent Claims 12-13 and 21

Claim 12 depends from claim 1 and additionally recites locally tracking within a selected member protocol progress information associated with all other members of a group. Claims 13 and 21 respectively depend from claims 1 and 14, and additionally recite locally tracking within a selected member protocol progress information associated with each other member of a group. In rejecting these claims, the Examiner yet again cites *Murphy*, col. 7, lines 1-20 (adding col. 6, lines 45-67 against claim 13). The cited passage, however, merely discloses the use of ACK messages to acknowledge INC and DEC messages. Furthermore, as discussed above, the Examiner has taken the position that client/server computers are analogous to "members" of a group, and Applicant fails to comprehend where in *Murphy* can be found any disclosure of tracking in one member protocol progress information associated with all other members or each other member of a group. If, for example, one node in the *Murphy* cluster has no object references, it does not appear that any information associated with that node is tracked by any other node, and indeed, that node does not participate in any messaging associated with the tracking of object reference counts. Accordingly, claims 12-13 and 21 are novel over *Murphy* and the rejections thereof should be reversed.

Dependent Claims 24 and 26

Claims 24 and 26 are not separately argued.

CONCLUSION

In conclusion, Applicant respectfully requests that the Board reverse the Examiner's rejections of claims 1-26, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324. Moreover, if any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

WOOD, HERRON & EVANS, L.L.P.

Date: February 2, 2007

2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
(513) 241-2324 - Telephone
(513) 241-6234 - Facsimile

By: /Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323

VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 09/732,189)

1. (Once Amended) A method of determining a status of a peer protocol initiated on a plurality of members of a group in a clustered computer system, the method comprising:
 - (a) locally tracking protocol progress information for a peer protocol within each of a plurality of members collectively managed as a group by a clustered computer system, wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol; and
 - (b) responding to a query directed to a selected member of the group by providing the protocol progress information locally tracked by the selected member, wherein the query comprises a request for the protocol progress information.
2. (Original) The method of claim 1, wherein locally tracking protocol progress information includes tracking, within a first member of the group, acknowledgment (ACK) messages directed to the first member by each other member of the group.
3. (Original) The method of claim 1, wherein locally tracking protocol progress information includes:
 - (a) tracking, within a first member of the group, a current acknowledgment (ACK) round for the first member, the current ACK round associated with a current peer protocol being processed by the first member; and
 - (b) tracking, within the first member, a last ACK round received parameter associated with each other member of the group, the last ACK round received parameter for each other member identifying a peer protocol associated with a last received ACK message from such other member.
4. (Original) The method of claim 3, wherein locally tracking protocol progress information further includes updating the current ACK round for the first member in response to receipt of ACK messages for the current peer protocol from all other members of the group.

5. (Original) The method of claim 1, wherein locally tracking protocol progress information includes updating the protocol progress information for a first member of the group in response to receipt of an acknowledgment (ACK) message directed to the first member.

6. (Original) The method of claim 1, further comprising:

(a) waiting on a resource required by a protocol being processed on the selected member; and

(b) monitoring for receipt of the query by the selected member while waiting on the resource.

7. (Original) The method of claim 6, wherein the protocol is a peer protocol, and wherein waiting on the resource includes waiting for receipt of an acknowledgment (ACK) message directed to the selected member.

8. (Original) The method of claim 6, wherein the protocol is a local protocol, and wherein waiting on the resource includes waiting on a local resource requested by the selected member.

9. (Original) The method of claim 8, wherein the local resource is selected from the group consisting of a lock and a creation of a new job.

10. (Original) The method of claim 6, wherein waiting on the resource includes waiting for receipt of a message by a local message queue for the selected member, and wherein monitoring for receipt of the query includes monitoring the local message queue for receipt of a query message.

11. (Original) The method of claim 1, wherein locally tracking protocol progress information within each member of the group includes locally tracking within the selected member protocol progress information associated with at least one other member in the group.

12. (Original) The method of claim 1, wherein locally tracking protocol progress information within each member of the group includes locally tracking within the selected member protocol progress information associated with all other members in the group.

13. (Original) The method of claim 1, wherein locally tracking protocol progress information within each member of the group includes locally tracking within each member protocol progress information associated with each other member in the group.

14. (Once Amended) An apparatus, comprising:

(a) a memory; and

(b) a program resident in the memory, the program configured to determine a status of a peer protocol initiated on a plurality of members that are collectively managed as a group by a clustered computer system by locally tracking protocol progress information within at least one member of the group, and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member, wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol, and wherein the query comprises a request for the protocol progress information.

15. (Original) The apparatus of claim 14, wherein the program is configured to locally track protocol progress information by tracking, within a first member of the group, acknowledgment (ACK) messages directed to the first member by each other member of the group.

16. (Original) The apparatus of claim 14, wherein the program is configured to locally track protocol progress information by tracking, within a first member of the group, a current acknowledgment (ACK) round for the first member, and tracking, within the first member, a last ACK round received parameter associated with each other member of the group, wherein the current ACK round is associated with a current peer protocol being processed by the first

member, and wherein the last ACK round received parameter for each other member identifies a peer protocol associated with a last received ACK message from such other member.

17. (Original) The apparatus of claim 14, wherein the program is further configured to wait on a resource required by a protocol being processed on the selected member, and monitor for receipt of the query by the selected member while waiting on the resource.

18. (Original) The apparatus of claim 17, wherein the protocol is a peer protocol, and wherein the program is configured to wait on the resource by waiting for receipt of an acknowledgment (ACK) message directed to the selected member.

19. (Original) The apparatus of claim 17, wherein the protocol is a local protocol, and wherein the program is configured to wait on the resource by waiting on a local resource requested by the selected member.

20. (Original) The apparatus of claim 17, wherein the program is configured to locally track protocol progress information by locally tracking within a first member protocol progress information associated with at least one other member in the group.

21. (Original) The apparatus of claim 17, wherein the program is configured to locally track protocol progress information by locally tracking within each member protocol progress information associated with each other member in the group.

22. (Twice Amended) A clustered computer system, comprising:

- (a) a plurality of nodes coupled to one another over a network;
- (b) a plurality of members collectively managed as a group by the clustered computer system and configured to be executed by at least one of the plurality of nodes; and

- (c) a program configured to be executed by at least one of the plurality of nodes to determine a status of a peer protocol initiated on the plurality of members by locally

tracking protocol progress information within at least one member of the group, and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member, wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol, and wherein the query comprises a request for the protocol progress information.

23. (Once Amended) A program product, comprising:

(a) a program configured to determine a status of a peer protocol initiated on a plurality of members that are collectively managed as a group by a clustered computer system by locally tracking protocol progress information within at least one member of the group, and providing the protocol progress information locally tracked by a member of the group in response to a query directed to such member, wherein the peer protocol is of the type wherein each member of the group receives a message associated with the peer protocol and returns an acknowledgment in association with locally processing the peer protocol, and wherein the query comprises a request for the protocol progress information; and

(b) a signal bearing medium bearing the program.

24. (Original) The program product of claim 23, wherein the signal bearing medium includes at least one of a recordable medium and a transmission medium.

25. (Once Amended) An apparatus, comprising:

(a) a memory; and

(b) a program, resident in the memory, the program configured to monitor for receipt of a query message that requests protocol status information by a member among a plurality of members that are collectively managed as a group by a clustered computer system while a current protocol for the member is waiting on a resource, the program further configured to output protocol status information in response to receipt of the query message.

26. (Original) The apparatus of claim 25, wherein the resource is selected from the group consisting of a local resource and an acknowledgment (ACK) message.

IX. EVIDENCE APPENDIX

09/732,189

None.

X. RELATED PROCEEDINGS APPENDIX

09/732,189

None.